

# Nicklaus Choo

SOFTWARE ENGINEER · GOOGLE

☎ (+1) 669-274-8810 | ✉ [nicklauscyc@gmail.com](mailto:nicklauscyc@gmail.com) | 📱 [nicklauscyc](#) | 🌐 [nicklaus-choo](#)

## Work Experience

**Google** SOFTWARE ENGINEER | SERVER LOAD BALANCING [SunnyVale, CA, USA](#)

Aug 2022 - Present

- Designed and Implemented load balancing for ML/server workloads using GPU/TPU/KV-cache utilization. This achieved:
  - 96% faster Time-to-First-Token P90 latency.
  - 60% faster Normalized-Time-Per-Output-Token P90 latency.
  - 32% faster overall throughput for prefix-heavy workloads.
- Led the design and implementation of a distributed load balancing client-server system with partner teams in Canada and Poland. This achieved:
  - Multi-threaded, concurrent, sharded, distributed load balancing client-server resilient to regional outages.
  - Cloud customers can balance service traffic along any utilization dimension.
  - Vastly simpler configuration (50% reduction in configuration programming effort.) for custom load balancing behavior.
- Automated end-to-end testing framework for 70% reduction in overall manual effort.
- 10x speedup for network design process with 15x reduction in memory resources for network topology graph data structures.
- Wrote graph traversal algorithms to efficiently traverse Google's network topology and detect single points of failure.
- Improved automation to re-map 25% of all edge network customers to greatly improve customer cost center allocation.

**NetApp** SOFTWARE ENGINEER INTERN [SunnyVale, CA, USA](#)

May 2021 - Aug 2021

- 3x speedup for OS compile-update-reboot time for ONTAP virtual machines.
- Automated ONTAP cloud cluster setup configuration for dynamic load testing.
- 2.7x speedup for WAFL scheduler client I/O latency with minimal slowdown (0.8x) to WAFL scheduler replication operations latency.
- Further optimized 3.5x speedup for client I/O and 1.2x speedup for replication operations with online random forest server load prediction.

**ZODAJ** FULL STACK SOFTWARE ENGINEER INTERN [Pittsburgh, PA, USA](#)

May 2020 - Aug 2020

- Devised SMS and Android COVID-19 contact tracing for Senegal in partnership with Senegalese health authorities.
- Designed NoSQL, PostgreSQL databases along with AWS Lambda RESTful API for TCN protocol contact tracing.
- Directed and implemented all permissions and roles within ZODAJ for Amazon AWS databases.
- Created exposure overlap algorithm for  $n$  people in a store that runs in  $\max(O(k) \text{ or } O(n \log n))$ , where  $k$  is overlap count.
- Wrote graph algorithms, discrete-time Markov chain SIR epidemic modeling for tracking infections.

## Noteworthy Projects

### 32-bit Kernel

- Created x86 kernel from scratch which supports essential syscalls such as fork/exec/wait, pre-emptive multitasking.
- Wrote device drivers for keyboard, timer, and hardware cursor.
- Code available at [github.com/nicklauscyc/small-kernel](https://github.com/nicklauscyc/small-kernel)

### Visual Question Answer

- Created a computer vision model to answer questions about images using a Convolutional Neural Network in pytorch
- Code available at [github.com/nicklauscyc/vqa](https://github.com/nicklauscyc/vqa)

## Education

### Carnegie Mellon University

[Pittsburgh, PA, USA](#)

B.S. IN COMPUTER SCIENCE, CONCENTRATION IN ALGORITHMS & COMPLEXITY WITH UNIVERSITY HONORS

Aug 2018 - May 2022

- School of Computer Science Dean's List, High Honors F19, F20, F21.
- Selected courses:

**10-701** Machine Learning (PhD)

**15-410** Operating Systems

**15-411** Compiler Design (C++)

**15-440** Distributed Systems (Java)

**15-451** Algorithm Design & Analysis

**15-459** Quantum Computation

**15-356** Cryptography

**15-210** Parallel Data Structures & Algos

**15-213** Introduction to Computer Systems

**15-251** Great Ideas in Theoretical CS

**15-259** Probability & Computing

**15-260** Statistics & Computing

**21-241** Linear Algebra

**21-259** Calculus 3D

**21-301** Combinatorics

**21-373** Abstract Algebra

**80-413** Category Theory

## Technical Proficiencies

### Languages

- C | C++ | Python | Go | Java | SML
- YAML | Bash | HTML | CSS | JavaScript

### Frameworks

**Google Kubernetes Engine** production server configuration

**Borg** production cluster turn-up and monitoring

**Docker** containerized OS build testing, app testing

**Terraform** infrastructure configuration

**Pod** release automation / monitoring / observability